

Nama : Khansa Alvita Amelia  
NPM : 50422790  
Kelas : 2IA11  
Matkul: Pemrograman Berbasis Objek

---

## Rangkuman Bab 1-6

Pemrograman berorientasi objek (Object Oriented Programming atau disingkat OOP) adalah paradigma pemrograman yang berorientasikan kepada objek yang merupakan suatu metode dalam pembuatan program, dengan tujuan untuk menyelesaikan kompleksnya berbagai masalah program yang terus meningkat. Objek adalah entitas yang memiliki atribut, karakter (bahaviour) dan kadang kala disertai **kondisi (state)**.

Pemrograman ini ditemukan pada Tahun 1960 ini berawal dari suatu pembuatan program yang terstruktur (structured programming). Metode ini dikembangkan dari bahasa C dan Pascal. Dengan pemrograman ini, menghasilkan penulisan program yang sulit menjadi lebih mudah.

Perbedaan diantara OOP dan Pemrograman terstruktur :

### OOP

- Penggabungan fungsi dan data dalam kelas-kelas atau objek-objek.
- Memiliki ciri dari Encapsulation, Inheritance, dan Polymorphism.
- Struktur program ringkas, cukup memerlukan object & class.
- Kode program re-usable

### Pemrograman Terstruktur

- Memecah program dalam fungsi dan data.
- Struktur program cukup rumit.
- Pengaplikasian reuse kode masih kurang.
- Efektif digunakan untuk masalah kecil hingga sedang.
- Eksekusi lebih lambat dari pada pemrograman berorientasi objek.

Class adalah cetak biru atau blueprint dari object. Class digunakan hanya untuk membuat kerangka dasar. Diibaratkan dengan kendaraan, memiliki ciri-ciri seperti merk, memiliki mesin, memiliki ban, dan beberapa ciri khas lain yang menyatakan sebuah benda tersebut adalah kendaraan. Sebuah kendaraan juga bisa dikenakan tindakan, seperti menghidupkan mesin atau mematikan mesin.

```
// Deklarasi Class
public class Kendaraan{
    // Ini Adalah Class, semua konstruktor, variabel, method berada disini
}
```

Suatu Objek adalah unik yang mewakili sesuatu di dunia nyata contohnya : orang, mobil dan lainnya. Objek mirip dengan suatu rekaman (record) dalam sebuah sistem berkas. Suatu objek didefinisikan berdasarkan namanya, objek juga memiliki atribut dan metode. Kesimpulannya Objek adalah entitas yang memiliki atribut, karakter dan kadang kala disertai kondisi.

Ada dua karakteristik yang utama pada sebuah objek , yaitu :

- Setiap objek memiliki atribut sebagai status yang kemudian akan disebut sebagai **state**.
- Setiap objek memiliki tingkah laku yang kemudian akan disebut sebagai **behaviour**.

Objek dalam perangkat lunak akan menyimpan **state-nya dalam variable** dan menyimpan informasi **tingkah laku (behaviour) dalam method atau fungsi-fungsi/prosedur**

Inheritance (pewarisan) adalah merupakan sebuah bentuk “penggunaan kembali” (*reusability*); dimana **class** baru dibuat dari **class** yang pernah ada yang (biasanya) ditambah fasilitasnya. Setiap **class** turunan dapat menjadi **class** pokok (induk) untuk **class** turunan yang akan datang. Dalam pewarisan, *constructor* tidak diwariskan pada **class** turunannya, kecuali jika digunakan perintah **super**.

Class yang mewariskan = superclass / parent class / base class.

Class yang mewarisi (class yang baru) = subclass / child class / derived class.

Penerapan inheritance menggunakan statement “**extends**”, Keyword “**super**” digunakan oleh subclass untuk memanggil constructor, atribut dan method pada superclassnya.

Memanggil constructor milik superclass

```
super ()  
super (parameter)
```

Memanggil atribut dan method milik superclass

```
super.namaAtribut  
super.namaMethod(parameter)
```

Enkapsulasi (encapsulation) merupakan cara untuk melindungi property (atribut) / method tertentu dari sebuah kelas agar tidak sembarangan diakses dan dimodifikasi oleh suatu bagian program. Cara untuk melindungi data yaitu dengan menggunakan access modifiers (hak akses). 4 hak akses yang tersedia, yaitu default, public, protected, private.

Overloading adalah diperbolehkannya sebuah class memiliki lebih dari satu nama function/method yang sama tetapi memiliki parameter/argument yang berbeda. Overriding

adalah kemampuan dari subclass untuk memodifikasi method dari superclass-nya, dengan cara menumpuk (mendefinisikan kembali) method superclass-nya.